Ahmet Faruk Çolak
22102104
2024-2025 Fall

EE-202 LAB 3 FINAL REPORT

1-Introduction

Lab 3 focuses on understanding filter design and implementation, crucial in various applications such as signal processing and electronics. We explore the effective use of resistors (R), inductors (L), and capacitors (C) by designing and analyzing both passive and active filters, highlighting their key differences. The experiment is divided into two stages. Stage one involves constructing a passive low-pass filter (LPF) with predetermined R, L, and C values at a specific center frequency, and determining the inductor's resistance, a major contributor to insertion loss. Stage two centers around designing an active band-pass filter using an operational amplifier (op-amp), presenting a design challenge due to the freedom in choosing resistor and capacitor values.

2-Analysis

A- Low Pass Filter (Passive)

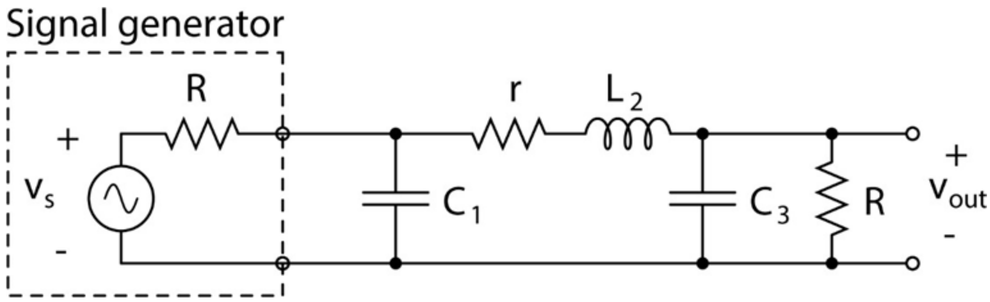

Figure.1: Low Pass Filter

How calculate required values:

$$C_1 = \frac{b_1}{2\pi f_0 R}, \quad L_2 = \frac{b_2 R}{2\pi f_0}, \quad C_3 = \frac{b_3}{2\pi f_0 R}, \qquad b_1 = 1.00 \quad b_2 = 2.00 \quad b_3 = 1.00$$

$$r = \frac{2\pi f L_{nom}}{Q} \quad L_{nom} = 15\mu H, \ \ Q = 25$$

Calculated values:

$ID: 22102104$

$f = 1065kHz$

$C1 = C3 = 2.7nF$

$L2 = 15\mu H$

$r = 10\ \Omega$

$R = 50\ \Omega$

B- Band Pass Filter (Active)

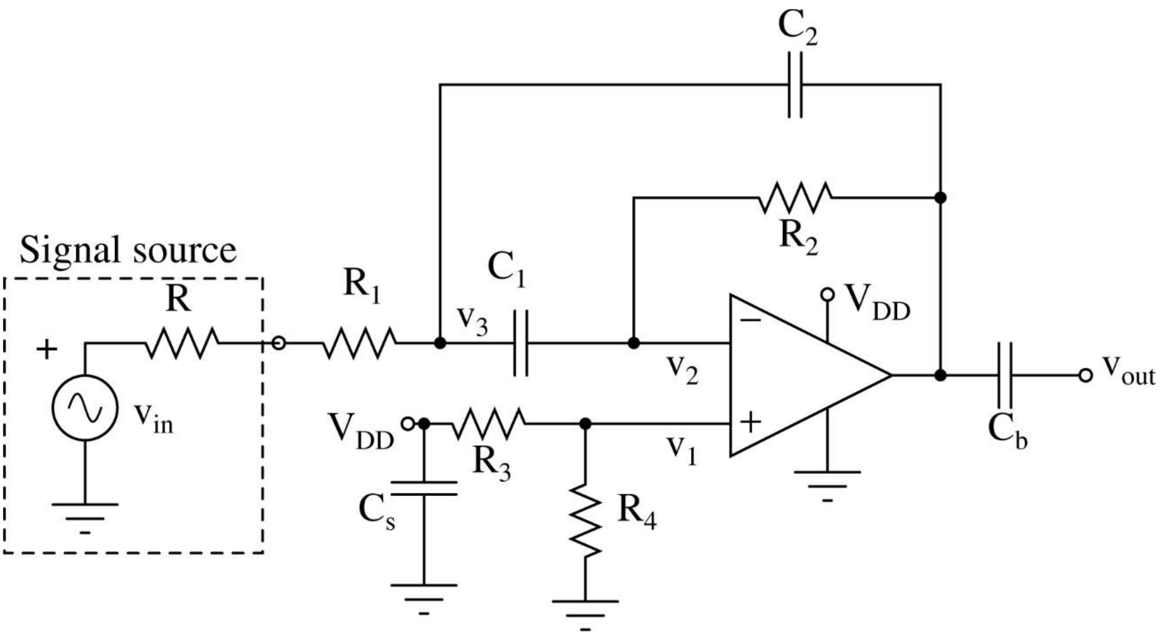

Figure.2: Band Pass Filter

Finding R and C values was a bit hard. That's why a python code was implemented to find consistent component values. The goal of the code is basically changing R and C values until finding fair match of R and C combination.

```
import math

R1 = 560
R2 = 39000
C1 = (1.2e-9)

Q=27/7

Val = (R2 / (4 * (R1 + 50))) ** 0.5

BW = 27/Val

Center_F = 1 / (2 * math.pi * C1 * (((R1 + 50) * R2)**0.5))

print("\nF_Center:", Center_F, "\n")

print("Şu an olan:", Val, "\n")
print("Olması Gereken", Q , "\n")

print("BW:", BW, "\n")

print("R1:", R1)
print("R2:", R2)
print("C1:", C1)
```

Figure.3: Python Code

$R_1 = 560\ \Omega$

$R_2 = 39000\ \Omega$

$C_1 = 1.2\ nF$

3-Preliminary Work

A- Low Pass Filter (Passive)

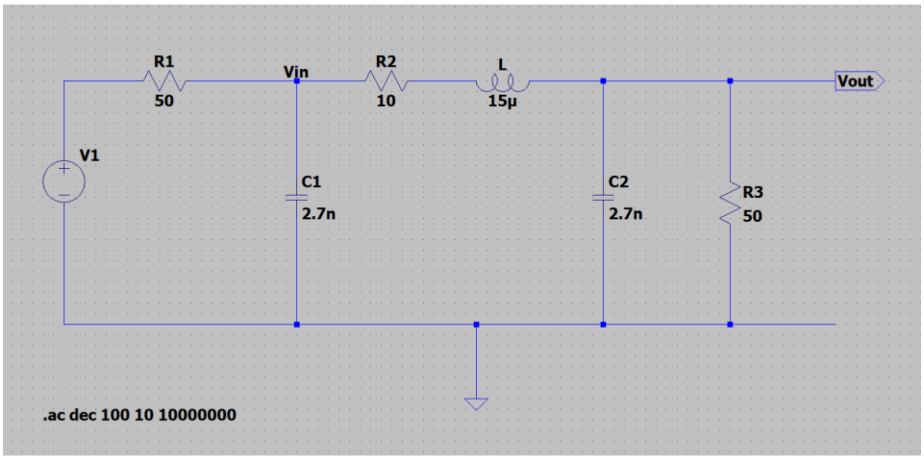

Figure.4: LTspice schematics for LPF

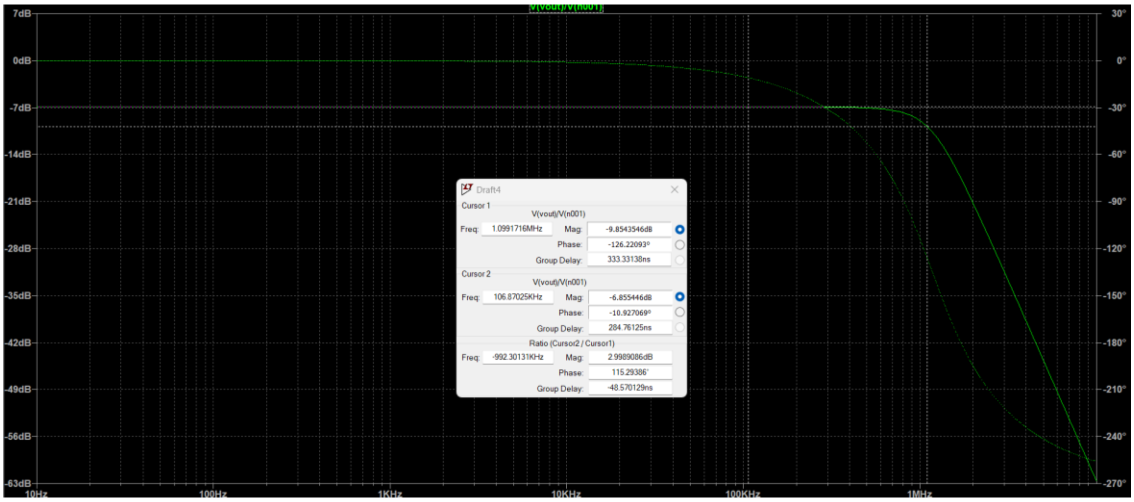

Figure.5: LTspice simulation for LPF

Cursor measurements revealed a 3dB cutoff frequency of 1099 KHz, which was 808 kHZ in preliminary lab work and an insertion loss of 6 dB. The observed band reject rate of 17.62 which was between-10dB/octave and -12dB/octave in preliminary work. The difference between these values is because of misunderstanding from mine. While I was doing simulation in LTspice, I should have got the signal before 50 Ω.

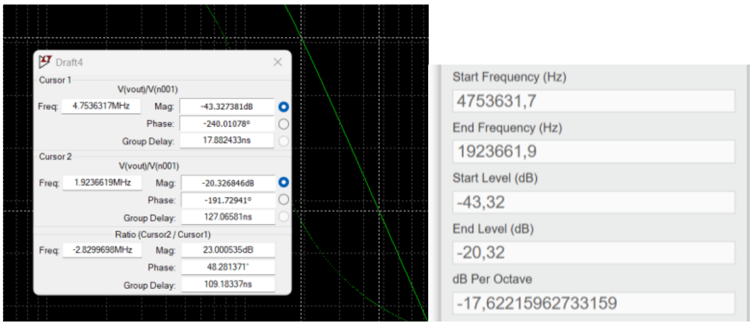

Figure.6: Band reject rate calculation for LPF

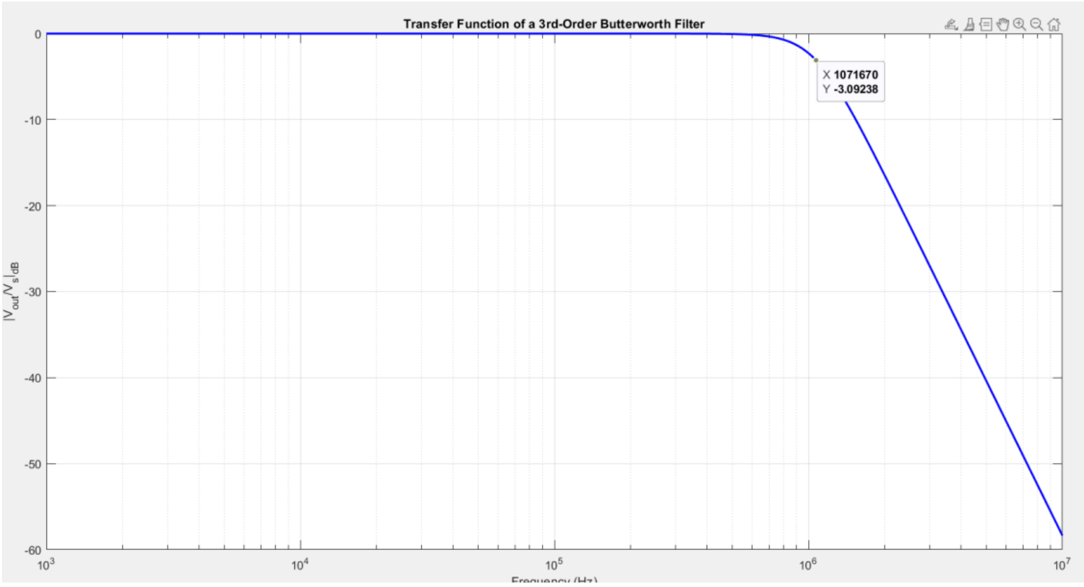

Figure.7: MATLAB simulation for LPF

According to simulation in MATLAB, f0 is approximately 1065K.

```
f0 = 1065e3;
n = 3;
frequencY = logspace(3, 7, 500);

H_dB = (-1)*10 * log10(1 + (frequencY / f0).^(2 * n));

figure;
semilogx(frequencies, H_dB, 'b-', 'LineWidth', 1.5);
xlabel('Frequency (Hz)');
ylabel('|V_{out}/V_s|_{dB}');
title('Transfer Function');
grid on;
```

Figure.8: MATLAB code for LPF

B- Band Pass Filter (Active)

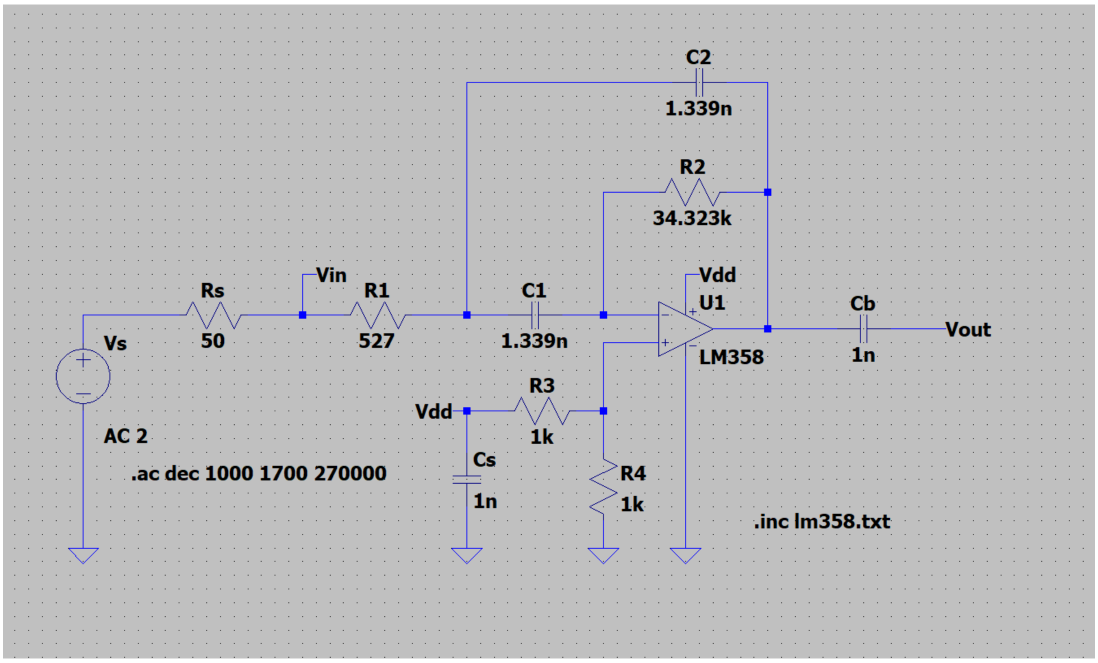

Figure.9: LTspice schematics of Band Pass Filter

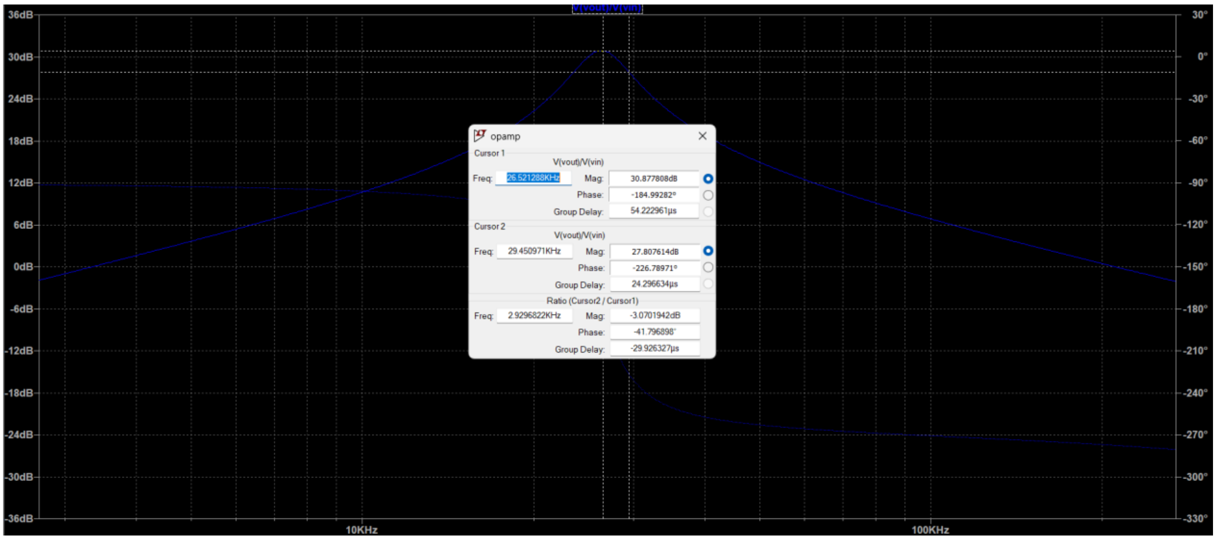

Figure.10: LTspice simulation of Band Pass Filter

LTspice simulation was updated after preliminary repot submission. Updated one is much more consistent with experimental results.

5-Hardware Implementation and Results

Methadology

Low-Pass Filter (LPF):

Standard formulas were used to calculate the cut-off frequency. However, available component values deviated from calculated values due to practical limitations and tolerances, potentially leading to minor variations in the results.

Band-Pass Filter (BPF):

Accurately achieving the target center frequency and bandwidth values provided was a bit challenging. To optimize the resistor (R) and capacitor (C) values and obtain more precise results, a Python program was employed. Initially, the following method was implemented to achieve greater precision, however, this approach did not yield the desired outcome due to various factors:

- **Resistors:** Resistors were connected in series to obtain the desired resistance values.
- **Capacitors:** Capacitors were connected in parallel to achieve the desired capacitance values.

A more straightforward method was then adopted, accepting a reasonable tolerance around the target center frequency and bandwidth. The Python code was revised to determine R and C values based on this more practical approach, resulting in a functional system.

Results

Low Pass Filter

	Frequency (Hz) 1-7	Vout (mV) 1-7	Frequency (Hz) 8-14	Vout (mV) 8-14
1	106500	2000	1704000	700
2	213000	1920	1917000	700
3	426000	1760	2130000	700
4	852000	1450	4260000	300
5	1065000	1280	8520000	260
6	1278000	1080	10650000	260
7	1491000	800	21300000	300

Figure.11: Frequency-Peak to peak voltage output of Low Pass Filter

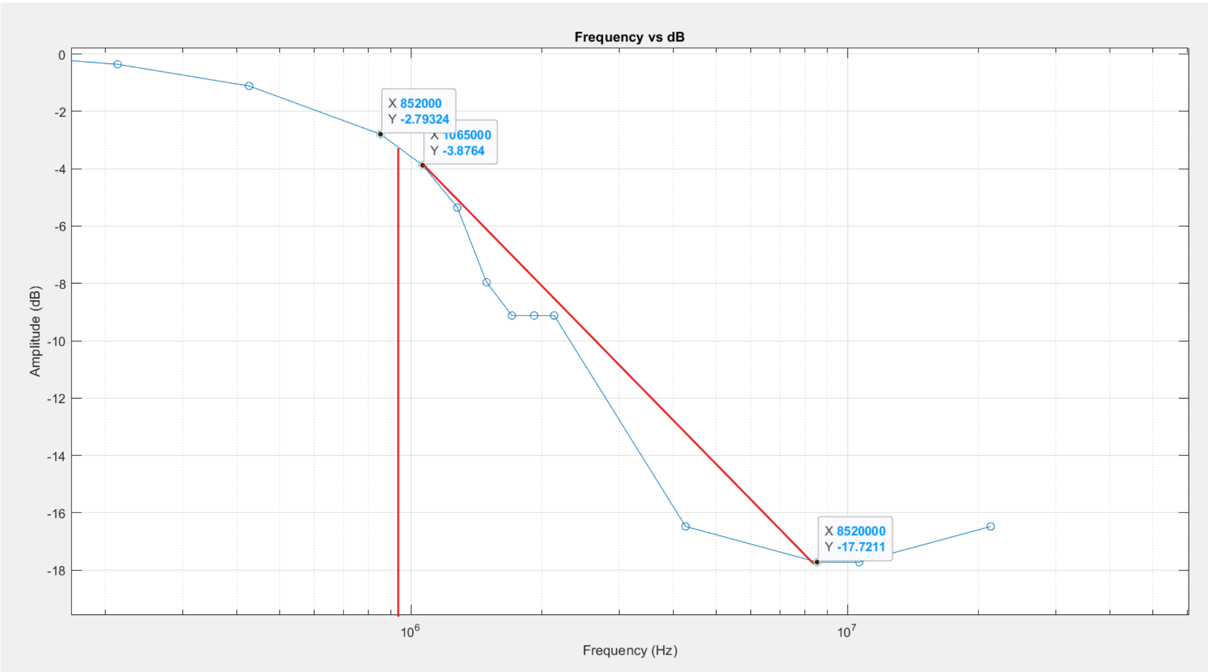

Figure.12: Hardware Implementation of LPF

Peak to peak voltages for every given frequency were noted and the above graph was implemented by MATLAB. 3dB cut-off frequency, ~980K, is slightly lower than 1065K, with 7.98% error, which is acceptable. Insertion loss was almost zero at 0.1fc.

```
freq = [106500, 213000, 426000, 852000, 1065000, 1278000, 1491000, 1704000, 1917000, 2130000,
4260000, 8520000, 10650000, 21300000];
Vin = [2000, 1920, 1760, 1450, 1280, 1080, 800, 700, 700, 700, 300, 260, 260, 300];

V_ratio = Vin / 2000;
dB = 20 * log10(V_ratio);

figure;
semilogx(freq, dB, '-o');
xlabel('Frequency (Hz)');
ylabel('Amplitude (dB)');
title('Frequency vs dB');
grid on;
```

Figure.13: Hardware Implementation MATLAB code of LPF

Low pass filter was successfully implemented. Theoretical and experimental results are given in the table below. Results are consistent with preliminary work.

	Theoretical (MATLAB)	Theoretical (LTspice)	Experimental
Maximum dB	0 dB	0 dB	~0 dB
3 dB cut-off frequency	~1065KHz	~1099KHz	~980KHz
Reject region	-18 dB/octave	-17.43 dB/octave	-7.39 dB/octave

Table.1: Theoretical and Experimental Results of Low Pass Filter

Band Pass Filter

	Frequency (Hz) 1-8	Vout (mV) 1-8	Frequency (Hz) 9-16	Vout (mV) 9-16
1	2650	80	29150	1100
2	5300	100	31800	860
3	10600	200	47700	380
4	21200	880	67625	260
5	23850	1180	132150	160
6	25175	1300	265000	120
7	26500	1320	530000	120
8	27820	1240	1060000	100

Figure.14: Frequency-Peak to peak voltage output of Band Pass Filter

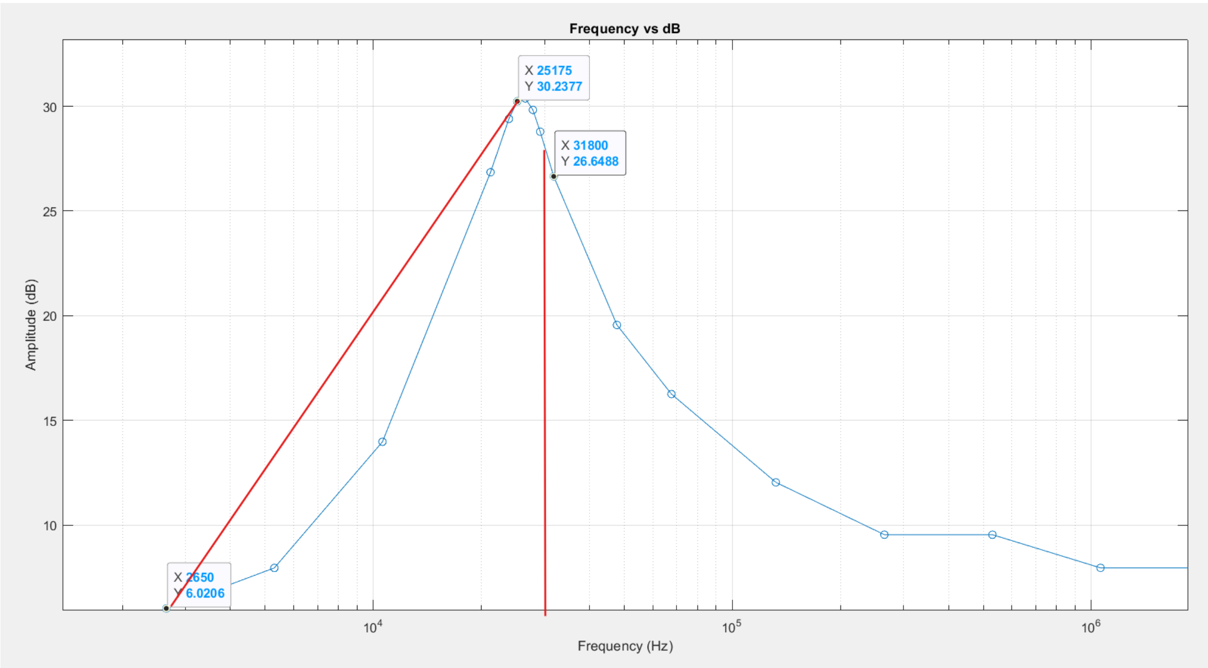

Figure.15: Hardware Implementation of Band Pass Filter

Same process was made for band pass filter. Center frequency, 26.5 kHZ, was almost same with theoretical processes. 3dB cut-off frequency, ~30K, is slightly lower than 29K, with 3.33% error, which is acceptable. Reject region is -7.45 dB/octave from top to bottom.

```
freq = [2650, 5300, 10600, 21200, 23850, 25175, 26500, 27820 29150, 31800, 47700, 67625, 132150,
265000, 530000, 1060000, 2120000];
Vin = [80, 100, 200, 880, 1180, 1300, 1320, 1240, 1100, 860, 380, 260, 160, 120, 120, 100, 100];

V_ratio = Vin / 40;
dB = 20 * log10(V_ratio);

% Frekans - dB plotunu çizme
figure;
semilogx(freq, dB, '-o');
xlabel('Frequency (Hz)');
ylabel('Amplitude (dB)');
title('Frequency vs dB');
grid on;
```

Figure.15: Hardware Implementation MATLAB code of LPF

Low pass filter was successfully implemented. Theoretical and experimental results are given in the table below. Results are consistent with preliminary work.

	Theoretical (LTspice)	Experimental
Maximum dB	30.87 dB	~30 dB
Center frequency	26.512 KHz	~26.5 KHz
3 dB cut-off frequency	29.45 KHz	~30 KHz
Bandwidth	~6.2 KHz	~7 KHz
Reject region	-7.47 dB/octave	-7.45 dB/octave

Table.2: Theoretical and Experimental Results of Band Pass Filter

5-Conclusion

This experiment focused on designing and implementing a Low Pass Filter (LPF) and a Band Pass Filter (BPF), providing hands-on experience in filter design, analysis, and optimization.

For the LPF, the provided formulas and LTspice simulations led to straightforward results. The experimental findings showed minimal insertion loss, which was expected due to the inductor's presence. The 3dB cutoff frequency was approximately 980 kHz, with a 7.98% error compared to the theoretical value of 1065 kHz. While the insertion loss was minimal, it cannot be completely disregarded due to slight fluctuations in measured peak-to-peak voltages.

The BPF design was more challenging. Initially, attempts to precisely match the given center frequency and bandwidth led to difficulties, partly due to an error in the Python code. After adjusting the approach to allow for approximate matching, the Python code was updated, simplifying the process. The manually measured components helped minimize deviations, resulting in a center frequency of ~26.5 kHz and a 3dB cutoff frequency of ~30 kHz, both aligning closely with theoretical values and having an acceptable error margin of 3.33%.

For the LPF, no special method was required, and the given formulas and LTspice outputs led smoothly to the results. However, the BPF required two approaches: the initial precise method, which proved inefficient, and the revised method with some tolerance, which ultimately succeeded.

This experiment provided a broad perspective on designing LPF and BPF circuits, reinforcing theoretical concepts with practical applications. Tools like Python simplified the design process and demonstrated the practical value of skills learned in other courses. This experiment highlighted the importance of adaptability and effective use of auxiliary tools in achieving desired circuit design outcomes.

Key Revelations

- Challenges in the BPF design were overcome by adjusting the approach and updating the Python code.

- Manual measurement of components helped minimize deviations and achieve accurate results.

- Using Python as a design aid showcased the practical utility of skills learned in other courses.